

臺北市九十九學年度高級中等學校電腦程式設計競賽決賽試題

(高中組)

說明：

1. 本試卷共有四題，每題 25 分。
 2. 請記得隨時備份自己的程式。
-

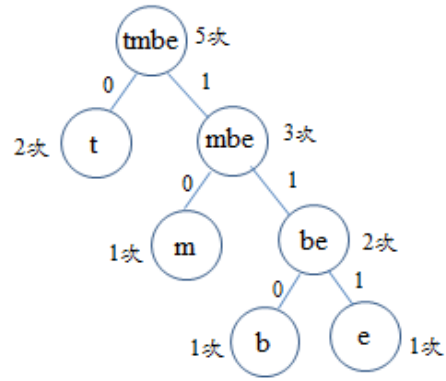
試題：

1. 數字間的秘密

對於一串(由 a 到 z 的英文字母組成)的英文字串，我們可將這行英文字串中的不同字母以下列敘述的方式轉換，變成由數字 0 與 1 所構成的序列：

- (i) 首先依英文字串中不同英文字母各自出現次數的多寡由小到大做排序(次數小的排在前面)；當次數相同時，則以英文字母的順序做為排序的依據(字母順序小的排在前面，例如 f 排在 m 之前)。
- (ii) 將排序最小的兩個字母合併成一個字母組(此字母組的出現次數設為原來兩個字母的出現次數和)並加入排序。此時，排序規則依然是先以次數排序，再以字母來排序。當字母組的次數與其它字母或字母組的次數相同時，字母組則將是以其所有字母中，由順序最小的字母做為此字母組參與排序之依據(例如 mbe 是會以 b 的字母順序最小，故 mbe 以 b 參與排序)。
- (iii) 注意在合併時可以是字母與字母、字母與字母組、字母組與字母組的合併。重複上述步驟直到將所有字母(組)合併完成。

以英文字串 bttme 為例，其出現的字母為 b、e、m、t，對應出現的次數分別為 1、1、1、2，執行上述步驟的過程可表示成下圖的樹狀結構，亦即 b 與 e 先合併成 be，m 再與 be 合併成 mbe，t 再與 mbe 最後合併成 tmbe。在圖中，每一個內部節點(代表某次合併後的字母組)會連到兩個子節點，分別代表合併前的兩個字母(組)，其中次數小的在左子節點；若兩個子節點次數相同時，則字母順序小的在左子節點。我們可把連到左子節點的邊標示為 0，連到右子節點的邊標示為 1。這樣一來，我們就可得到原來英文字串以數字 0 與 1 表示的結果。譬如，b 表示成 110、e 表示成 111、m 表示成 10、t 表示成 0；而原來英文字串可表示成 1100010111，其中有 4 個 0、6 個 1。



輸入說明

第一行為一串英文字串(由 a 到 z 的英文字母組成)。

輸出說明

請由螢幕印出兩個整數值，分別代表轉換成數字 0 與 1 所構成的序列後，其中 0 與 1 的數目。

輸入範例一

rook

輸出範例一

3 3

輸入範例二

bttle

輸出範例二

4 6

2. 比克的煩惱

那美克星球上居住著四個種族的生物，神族、蟲族、龍族和人類。不管是哪一個種族的生物，十八歲之後就要開始工作。星球上的職業分成五類，科學、生產、運動、飲食和服務。每一種職業注重的能力不同，科學家需要高度的創造力，生產者不可或缺的是源源不絕的力量，運動員靠的是敏捷度，飲食類以烹調力取勝，而服務業當然要有好口才。

那美克星球是一個階級分明的星球，用餐時一定要由階級高的生物優先進行，階級低的生物必須等階級高的生物用餐完畢才可以開始。對不同種族但從事同一個職業的生物而言，該職業注重的能力越高者，其階級就越高。例如兩個擔任科學家的生物，創造力低的要等創造力高的生物用餐後始可用餐。那美克星球也是一個長幼有序的星球，對同一種族而言，不論其職業為何，年紀越長的生物其階級越高。不同種族的生物不會以年紀作階級區分，不同種族且不同職業的生物之間沒有任何階級之分。

比克在那美克星球上開了一家餐廳，這間餐廳只接受預約用餐。比克會依據預約名單上客人的職業、年紀和各種能力值來安排客人用餐的順序。(假設客人之間的階級不會形成循環矛盾的情況。)每位客人的用餐時間固定為一小時。今天晚上預約的客人很多，但是比克的老婆希望他早點回家。他把預約名單交給了你，希望你幫他計算今天晚上最快要幾個小時才能讓所有客人用完餐，好讓他可以早早休息回家。

輸入格式

測試資料共有五筆，分別放在檔案 Bick1.in, Bick2.in, Bick3.in, Bick4.in 和 Bick5.in 中。每個檔案的第一行代表的是預約客人的人數 N 。第二行到第 $N+1$ 行，每一行代表一位客人的資料，依序是種族、職業、年紀、創造力、力量、敏捷度、烹調力和口才。每個欄位之間以一個空白 (space) 隔開。

種族代碼：神族(P)、蟲族(Z)、龍族(D)、人類(T)。

職業代碼：科學(S)、生產(F)、運動(A)、飲食(L)、服務(V)。

年紀數值範圍為 18~100 (皆已有職業)。

各種屬性值的數值範圍都在 1~10,000 之間。

輸出格式

請輸出最快要幾個小時才能在符合階級制度的情況下，讓所有客人用餐完畢。

輸入範例一

5

PA 20 10 10 10 10 10

P S 30 10 10 10 10 10
P S 40 30 20 20 20 20
Z S 22 25 19 18 17 16
D S 70 20 30 40 50 60

輸出範例一

5

(第三位客人 → 第四位客人 → 第五位客人 → 第二位客人 → 第一位客人，依此階級順序最快需要 5 個小時才能讓所有客人用餐完畢。)

輸入範例二

5

PA 40 10 10 10 10 10
P S 30 10 10 10 10 10
P V 20 30 20 20 20 60
Z F 50 25 19 18 17 16
Z V 40 20 30 40 50 30

輸出範例二

4

(第一位客人 → 第二位客人 → 第三位客人 → 第五位客人，依此階級順序最快需要 4 個小時才能讓所有客人用餐完畢。)

3. 訊息編碼與解碼

本題中你需要寫一個**編碼程式**以及一個**解碼程式**，編碼程式將訊息檔案內的文字資料轉換為編碼檔案內的資料，解碼程式則反向將編碼檔案內的資料轉換為訊息檔案內的文字資料。

訊息檔案資料範例如右圖，由可以印出的 ASCII 文字以及換列字元構成（圖中 代表空格字元，`\n`代表換列字元）。

訊息檔案範例

```
Wiiikiii was \n
the first.
```

對應的**編碼檔案**資料範例如右圖，以一空白列分隔分為兩部分，第一部份為編碼表，

編碼表

```
i stwfefrk\n
aW.
```

空白列

編碼檔案範例

```
i stwfefrk\n
aW.\n
\n
0001\n
0001 1111\n
1\n
0 0 0 1\n
001\n
101 111\n
10001010011001\n
00011100010000\n
11110101001100\n
11101110110110\n
01011010111010\n
01100100111110\n
10011001110101\n
10110001001011\n
110000000
```

亦即所有在該訊息中出現的字元（包括換列字元），依照出現次數排序（出現次數相同者順序隨意）；

空白列之後為編碼資料

編碼資料

```
0001\n
0001 1111\n
...
```

編碼規則如下：

1. 將編碼表中字元逐一對應至下列 120 組長度不同的 0/1 字串：0, 00, 01, 10, 000, 001, 010, ..., 110, 0000, 0001, ..., 1110, ..., 000000, 000001, ..., 111110，例如上述範例僅使用其中前 14 組，字元 `i` 的編碼為 0，字元 `s` 的編碼為 00，字元 `t` 的編碼為 01, ...
2. 將訊息檔案中字元逐一根據編碼表編碼，為了能夠解譯回來，輸出每一字元對應的編碼(假設長度為 n)前需要額外輸出其編碼字串的長度 (以 $n-1$ 個 0 加上一個 1 表示)，輸出每一字元對應的編碼後需要輸出 n 個 1 的結束字串，例如上例中字元 `W` 對應的編碼為 0001，則應該輸出 0001 0001 1111。如果訊息中連續幾個字母的編碼長度相同則僅需輸出一次長度字串以及結束字串，例如上例中字元序列 `iii` 編碼為 1 0 0 0 1，字元序列 `st` 編碼為 01 01 10 11，其中第一組 01 代表接下來是長度為 2 的編碼，第二組 01 代表 `s`，10 代表 `t`，11 則代表此一長度 2 的編碼序列結束。全部編碼完畢後請加上 0000000 的結束字串。編碼資料中所有空格以及換行字元純粹是排版及解釋用，解碼程式應該將其丟棄。

請將測試資料 `coding.zip` 解開至 `coding` 資料夾

其中包括上述範例的訊息檔案 `sample.text.txt` 以及編碼檔案 `sample.code.txt`，你也會在資料夾中看到另一測試訊息檔案 `plain1.txt`，以及其編碼檔案 `plain1.code.txt`。

`tools.zip` 中 `diff.ex_` 以及 `sum.ex_` 為兩個簡易的 win32 工具程式，請更名為 `diff.exe` 及 `sum.exe` (linux 系統中和內建的 `diff` 及 `sum` 相同)讓你檢查編碼與解碼是否正確，例如 `diff plain1.txt plain1.decode.txt` 可以比較 `plain1.txt` 和 `plain1.decode.txt` 兩個檔案內容的差異，`sum plain1.txt` 則可以得到 CRC32 的檢查碼。

請注意：

1. 你的解碼程式解出 `plain1.code.txt` 必須和 `plain1.txt` 完全一樣，你的編碼程式編碼 `plain1.txt` 所得到的編碼檔案不見得要和 `plain1.code.txt` 完全一樣，但是再解碼後必須和 `plain1.txt` 完全一樣。
2. 解碼與編碼程式分別佔 10 及 15 分。
3. 資料夾中還有一個 `testdata.zip` 檔，其中為密碼保護的測試資料，評分時會用到。

4. 旋轉木馬

有一個旋轉木馬遊戲台，其設計與運轉規則如下：在一個矩形區域的平面上，有一隻木馬，其停留位置之 x 座標與 y 座標都是整數(即它會停在如座標值為 $x=10, y=20$ 這樣的位置，但不會停在座標值為 $x=9.9, y=20.3$ 這樣的位置)，且其位置會顯示出來；以此矩形區域的中央點為圓心，這木馬會逆時針繞著圓心旋轉，玩這個遊戲的人可以設定其旋轉的速度 S (幾分鐘一轉)，同時也可以設定其旋轉的時間 T (幾秒)(即旋轉幾秒後停下來)。遊戲的玩法很簡單：每次都可以設定 S 與 T 的值 (S 與 T 都是整數值)，然後猜測木馬旋轉後會停留的位置(整數座標值)，在該位置上作記號，之後按鈕啟動旋轉木馬，若木馬停留在該作記號的位置上，則可以獲得獎品。(注意：根據設定的 S 、 T 值所得出來的座標值，四捨五入取整數後才是木馬會停下來的位置)。

王小明在課堂上學過相關的問題，知道可以運用旋轉矩陣運算來求得木馬預期停下來的位置座標，請你幫他寫個程式來達到這個目的，這樣他就可以不用靠運氣，可以正確預測並拿到獎品。為了簡化一些細節，假設這個平台是個 $100*100$ 的正方形區域， x 座標範圍是 $-100\sim 100$ ， y 座標範圍也是 $-100\sim 100$ ，程式的輸出輸入規格如下：

輸入：第一列為木馬起始位置之 x 座標與 y 座標值；第二列為 S 的值；
第三列為 T 的值。(注意： x, y, S, T 都是整數， S 的單位是分， T 的單位是秒)

輸出：木馬旋轉後停下來之位置的 x 座標與 y 座標值(四捨五入後取整數)

範例 1：

輸入：

50, 0

10

150

輸出：

0, 50

範例 2：

輸入：

80, 80

10

300

輸出：

-80, -80

提示：旋轉矩陣為 $M(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$